

# Programmazione I e Laboratorio

## Test Prova Pratica di laboratorio

(Tempo a disposizione - 2h)

Dato lo scheletro di codice qui riportato, lo si completi aggiungendo l'implementazione delle funzioni richieste.

bozza.c

```
#include <stdio.h>
#include <stdlib.h>

//List structure:
struct El {
    int info;
    struct El *next;
};

typedef struct El ElementoLista;

// Functions/Procedure to be implemented:

ElementoLista* readList(ElementoLista **lista, int dis);
int maxDistance(ElementoLista *lista);
ElementoLista* CancellaMax(ElementoLista **lista, int maxdis);

//Function to print all the elements of the list:
void printList(ElementoLista *list) {
    printf("(");
    while (list != NULL) {
        printf("%d ", list->info);
        list = list->next;
    }
    printf(")\n");
}

int main() {
    ElementoLista *list = NULL, *list_aux = NULL;
    int boundis, maxdis;
```

```

//Read the bounded distance
scanf("%d", &boundis);

//Read and print the list
list_aux = readList(&list, boundis);
printf("La lista bounded-%d e':\n", boundis);
if (list_aux != NULL)
{
    list = list_aux;
    list_aux = NULL;
}
printList(list);

//Compute and print the maxdistance of the list
maxdis = maxDistance(list);
printf("La distanza massima e':\n%d\n", maxdis);

//Removes the elements at distance maxdistance
list_aux = CancellaMax(&list, maxdis);
printf("La lista modificata e':\n");
if (list_aux != NULL) list = list_aux;
printList(list);

return 0;
}

```

Le funzioni da implementare devono rispettare le seguenti specifiche:

- **readList**: Chiamiamo distanza tra due interi  $n_1$  e  $n_2$  il valore assoluto della loro differenza ( $|n_1 - n_2|$ ). Dato un limite per la distanza passato alla funzione nel parametro formale *boundis* (che sarà un intero positivo), **readList** legge una sequenza di numeri interi positivi e termina l'acquisizione quando la distanza tra il numero letto e il successivo è strettamente maggiore del limite. I numeri letti devono essere memorizzati nell'ordine di acquisizione in una lista: l'ultimo numero letto (che fa terminare l'acquisizione) non va inserito nella lista. Se il valore limite della distanza (*boundis*) passato a **readList** è 3, la sequenza in input 1 4 7 2 deve creare una lista contenente nell'ordine (1, 4, 7).
- **maxDistance**: Questa funzione prende una lista e deve restituire la massima distanza tra due elementi consecutivi della lista. Per esempio, supponendo che la lista sia (5, 8, 15, 10, 12), la funzione **maxDistance** dovrà restituire 7.
- **CancellaMax**: dato il valore della distanza massima di una lista, **CancellaMax** deve cancellare il *primo* elemento di *tutte* le coppie della lista di partenza che sono a distanza massima. Per esempio, se la lista è (5, 8, 15, 10, 12), dato che in questo caso **CancellaMax** verrà chiamata con maxdis=7, la lista risultante dovrà essere (5, 15, 10, 12).

## Esempio

### Input

5  
2  
1  
4  
7  
4  
1  
3  
9

### Output

La lista bounded-5 e':  
(2 1 4 7 4 1 3 )  
La distanza massima e':  
3  
La lista modificata e':  
(2 1 3 )